

Υλοποίηση Δομής Γράφου

Για να αναπαραστήσουμε ένα γράφο, χρησιμοποιήσαμε έναν τύπο λίστας γειτνίασης. Παρόλα αυτά στις λίστες του πίνακα κατακερματισμού δεν τοποθετούνται γείτονες, αλλά κόμβοι με βάση την ταυτότητα τους. Επιπρόσθετα ο κάθε κόμβος αυτής της λίστας περιέχει ένα ArrayList που τοποθετεί μέσα με ποιους γειτονεύει (με βάση την απόσταση που διαβάζει ο γράφος). Ο αρχικός πίνακας κατακερματισμού αρχικοποιείται με 5 λίστες, και μόλις μία από τις λίστες έχει μέγεθος 20, ξανά αρχικοποιείται ο πίνακας με $5 * \text{το προηγούμενο του ύψος}$.

Όπως βλέπουμε στην φωτογραφία η πρώτη εκτύπωση είναι το τι περιέχουν η λίστες του πίνακα κατακερματισμού

Η δεύτερη εκτύπωση περιλαμβάνει το Array List του καθένα από τα πάραπάνω node των λιστών, που περιέχουν τους γείτονες τους.

```
g Print Hashtable
0:
1: <ID: 821>
2: <ID: 02>    <ID: 12>
3: <ID: 123>   <ID: 353>
4:

g Print Adjacency List
<ID: 821>: <ID: 02>    <ID: 12>    <ID: 123>    <ID: 353>
<ID: 02>: <ID: 123>    <ID: 353>    <ID: 12>    <ID: 821>
<ID: 12>: <ID: 02>    <ID: 123>    <ID: 353>    <ID: 821>
<ID: 123>: <ID: 353>    <ID: 02>    <ID: 12>    <ID: 821>
<ID: 353>: <ID: 123>    <ID: 02>    <ID: 12>    <ID: 821>
```

Αυτή η υλοποίηση μας είναι πιο αποτελεσματική όταν η συνάρτηση κατακερματισμού μας επιστρέφει το ίδιο αποτέλεσμα για πολλούς κόμβους. Ένα αρνητικό της παραπάνω υλοποίησης όμως, είναι ότι χρειαζόμαστε περισσότερο χρόνο για να δούμε αν γειτονεύουν δύο κόμβοι αφού πρώτα πρέπει να βρούμε τον κόμβο στην λίστα και στην συνέχεια να κάνουμε αναζήτηση στην λίστα γειτνίασης του.

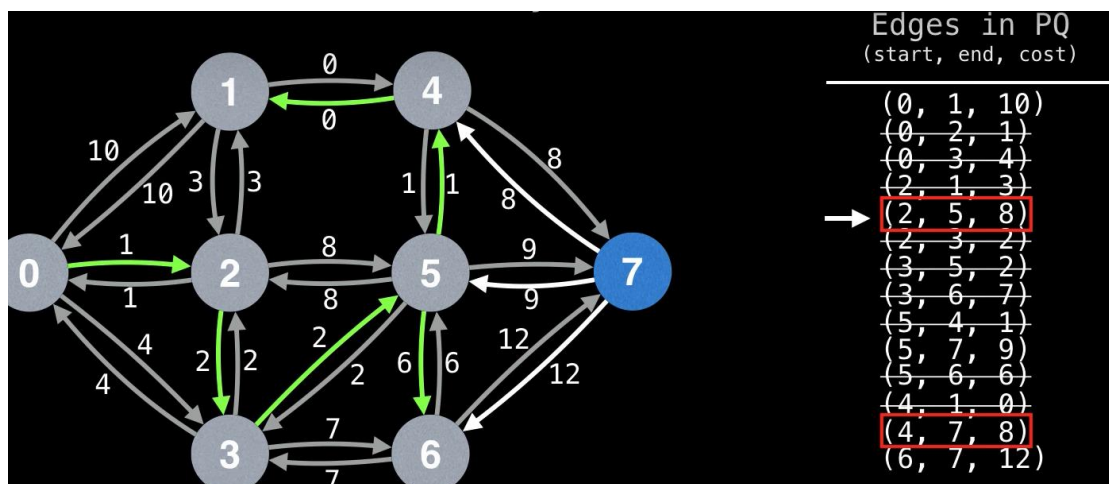
Υπολογισμός Minimum Spanning Tree

Αρχικοποιούμε και διατηρούμε Ουρά Προτεραιότητας που ταξινομεί τις ακμές με βάση το βάρος τους. Αυτό χρησιμοποιείται για να καθορίσω τον επόμενο κόμβο και την ακμή που μας πάει σε αυτόν.

Αρχίζουμε τον αλγόριθμο από τον κόμβο με ταυτότητα «02». Επισημαίνουμε ότι τον επισκεφθήκαμε και διασχίζει όλες τις ακμές του κόμβου, τοποθετώντας καθεμιά στην ουρά.

Καθώς η ουρά δεν είναι άδεια και δεν έχει δημιουργηθεί mst, κάνω dequeue την ακμή με το μικρότερο κόστος. Αν το τέλος της ακμής που έχω, έχει ήδη επισκεφθεί τότε την αγνοώ και ξανακάνω dequeue. Αλλιώς επαναλαμβάνω την διαδικασία παραπάνω.

Με το να ελέγχω ότι έχω επισκεφθεί τον κόμβο σιγουρεύω ότι δεν δημιουργούνται κύκλοι στο δέντρο μου.



Εκτύπωση MST και υπολογισμός μέγιστης θερμοκρασίας

Για τις παραπάνω λειτουργίες υλοποιήσαμε την διάσχιση BFS. Η διάσχιση BFS κάνω κατά πλάτος προσπέλαση του γράφου. Παραθέτουμε τον ψευδοκώδικα παρακάτω

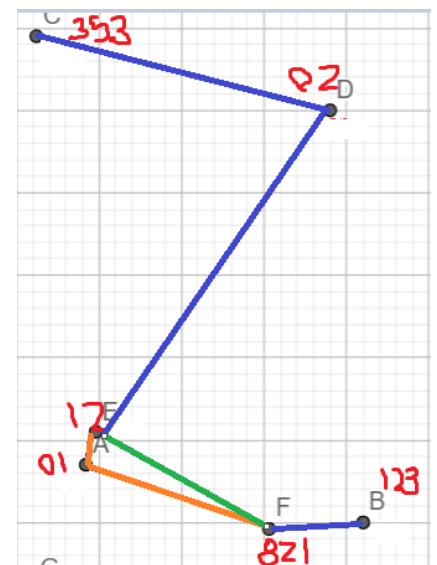
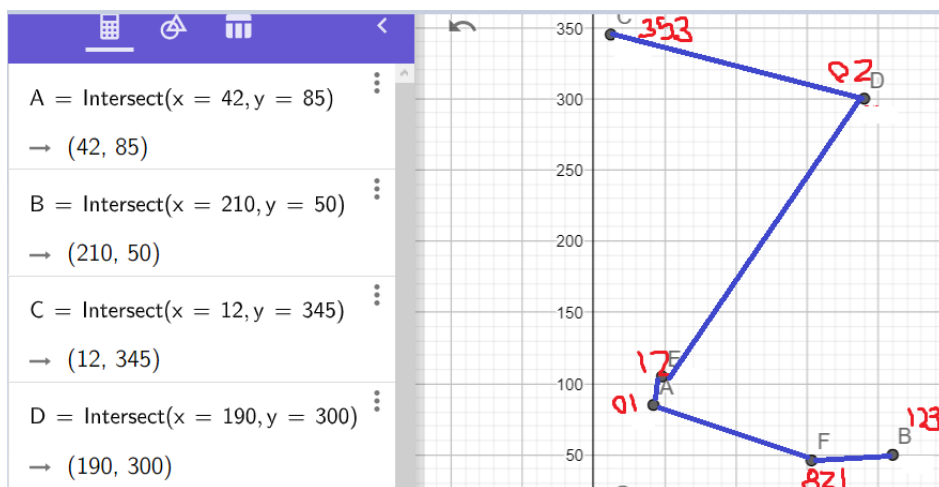
```
1  procedure BFS(G, root) is
2      let Q be a queue
3      label root as discovered
4      Q.enqueue(root)
5      while Q is not empty do
6          v := Q.dequeue()
7          if v is the goal then
8              return v
9          for all edges from v to w in G.adjacentEdges(v) do
10             if w is not labeled as discovered then
11                 label w as discovered
12                 Q.enqueue(w)
```

Βελτιστοποίηση μεθόδων μενού 3 & 4

Όπως μας ζητήθηκε οι διαδικασίες αυτή θα πρέπει να κάνουν τις απαραίτητες αλλαγές στο ελάχιστο γεννητορικό δένδρο που υπάρχει αποθηκευμένο. Η διαδικασία επαναυπολογισμού του ελάχιστου γεννητορικού δέντρου θα πρέπει να είναι όσον το δυνατόν γρηγορότερη (άρα πιθανών να μην πρέπει να χρησιμοποιήσετε τη μέθοδο του μενού 1, αλλά το γεγονός ότι ήδη έχετε το γεννητορικό δέντρο από προηγουμένως).

Η κλάση οδηγού πρόκειται να διατηρήσει δύο γράφους. Το αρχικό γράφο που οι κόμβοι είναι συνδεδεμένα με τους γείτονες τους βάση απόστασης, και ένα δευτερεύον γράφο που είναι το MST, αλλά αντιπροσωπεύεται ως γράφημα. Με αυτό τον τρόπο για την επιλογή δύο του μενού καλούμε μόνο BFS εκτύπωσης στο δευτερεύον γράφημα. Επίσης, για να προσθέσετε και να αφαιρέσετε έναν κόμβο γίνεται και στα δύο γραφήματα, έτσι ώστε όταν ο υπολογισμός γίνεται (στο δευτερεύον γράφημα), είναι ταχύτερη από τον υπολογισμό στο αρχικό γράφημα.

Για την διαγραφή ενός κόμβου: Διαγράφω τον κόμβο από το mst. Μετά συνδέω όλους τους γείτονες του κόμβου μεταξύ τους. Στην συνέχεια εκτελώ τον υπολογισμό mst στο ήδη υπάρχων mst. Δείτε παράδειγμα παρακάτω. Χρήση λογισμικού geogebra για αναπαράσταση του γράφου



Για την προσθήκη ενός κόμβου: Προσθέτω τον κόμβο από το mst. Στην συνέχεια συνδέω όσους κόμβους είναι στην μέγιστη απόσταση από τον συνδεδεμένο κόμβο. Μετά εκτελώ τον υπολογισμό mst στο ήδη υπάρχων mst. Δείτε παράδειγμα παρακάτω. Χρήση λογισμικού geogebra για αναπαράσταση του γράφου

